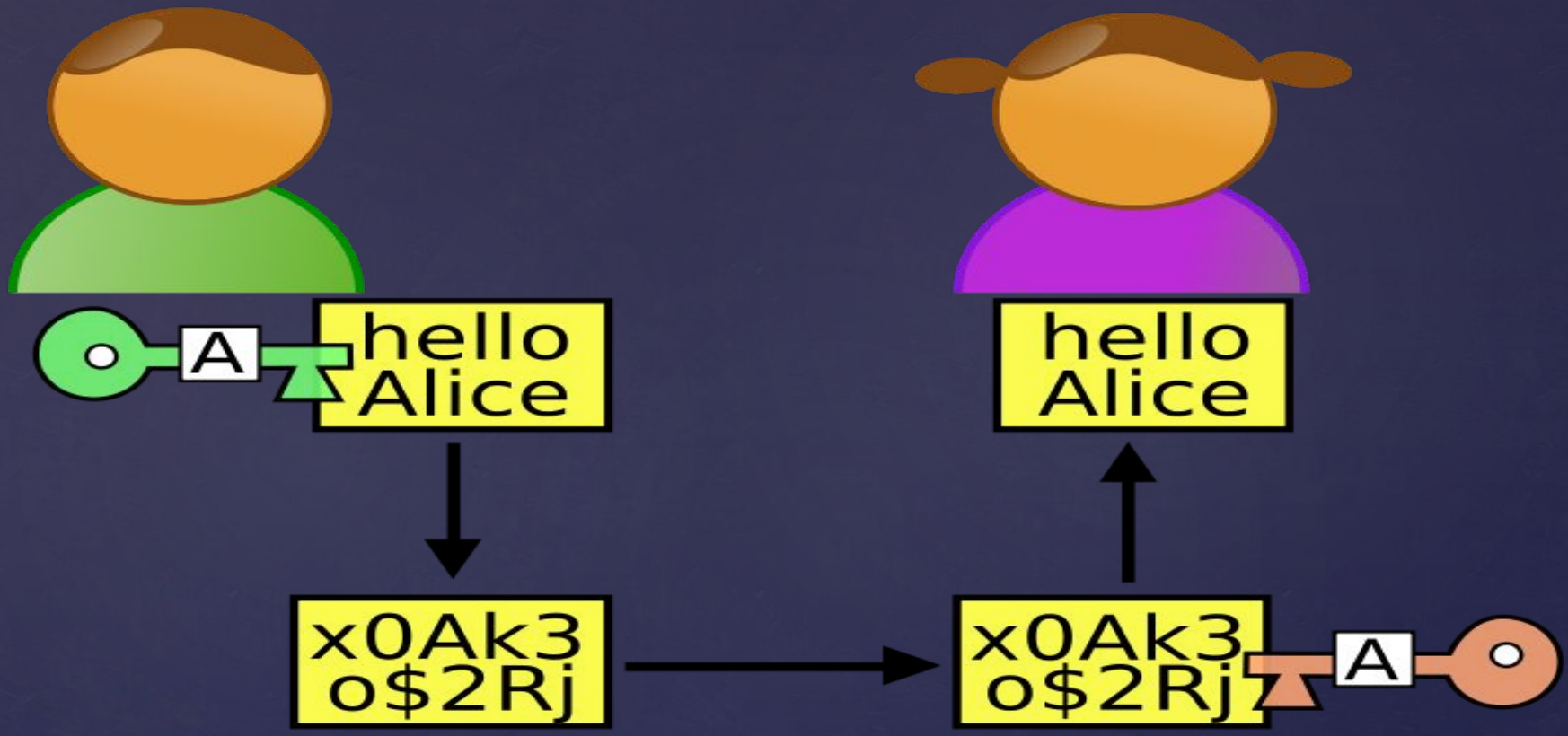
Unit-3 public key cryptography

{

- It is the study of techniques for ensuring the secrecy and/or authenticity of information. It has two main branches
- *Cryptography,*
  - Which is the study of the design of such techniques
- *Cryptanalysis,*
  - which deals with the defeating such techniques, to recover information that will be accepted as authentic

What is Cryptography???


A
hello
Alice
hello
Alice
x0Ak3
o$2Rj
x0Ak3
o$2Rj
A

Two main Types

- Symmetric Key Cryptography
- Asymmetric Key Cryptography (Public Key Cryptography)

Public key Cryptography

Why Public-Key Cryptography?

- developed to address two key issues:
  - **key distribution**
    - how to have secure communications in general without having to trust a KDC with your key
  - **digital signatures**
    - how to verify a message comes intact from the claimed sender

Principles of Public key Cryptography

- Asymmetric encryption is a form of cryptosystem in which encryption and decryption are performed using the different keys- one public key and one private key
- Also known as public-key encryption
- It uses mathematical functions rather than substitution and permutation
- More secure from cryptanalysis than the symmetric encryption

- **Asymmetric keys**
- Two related keys, a public key and a private key, that are used to perform complementary operations, such as encryption and decryption or signature generation and signature verification
- **Public key certificate**
- A digital document issued and digitally signed by the private key of a Certification authority
- It binds the name of a subscriber to a public key.
- The certificate indicates that the subscriber identified in the certificate has sole control and access to the corresponding private key

- **Public key cryptographic algorithm**
- A cryptographic algorithm that uses two related keys, a public key and a private key
- **Public key infrastructure**
- A set of policies, processes, server platform, software and workstations used for the purpose of administrating certificates and public-private key pairs, including the ability to issue, maintain, and revoke public certificate

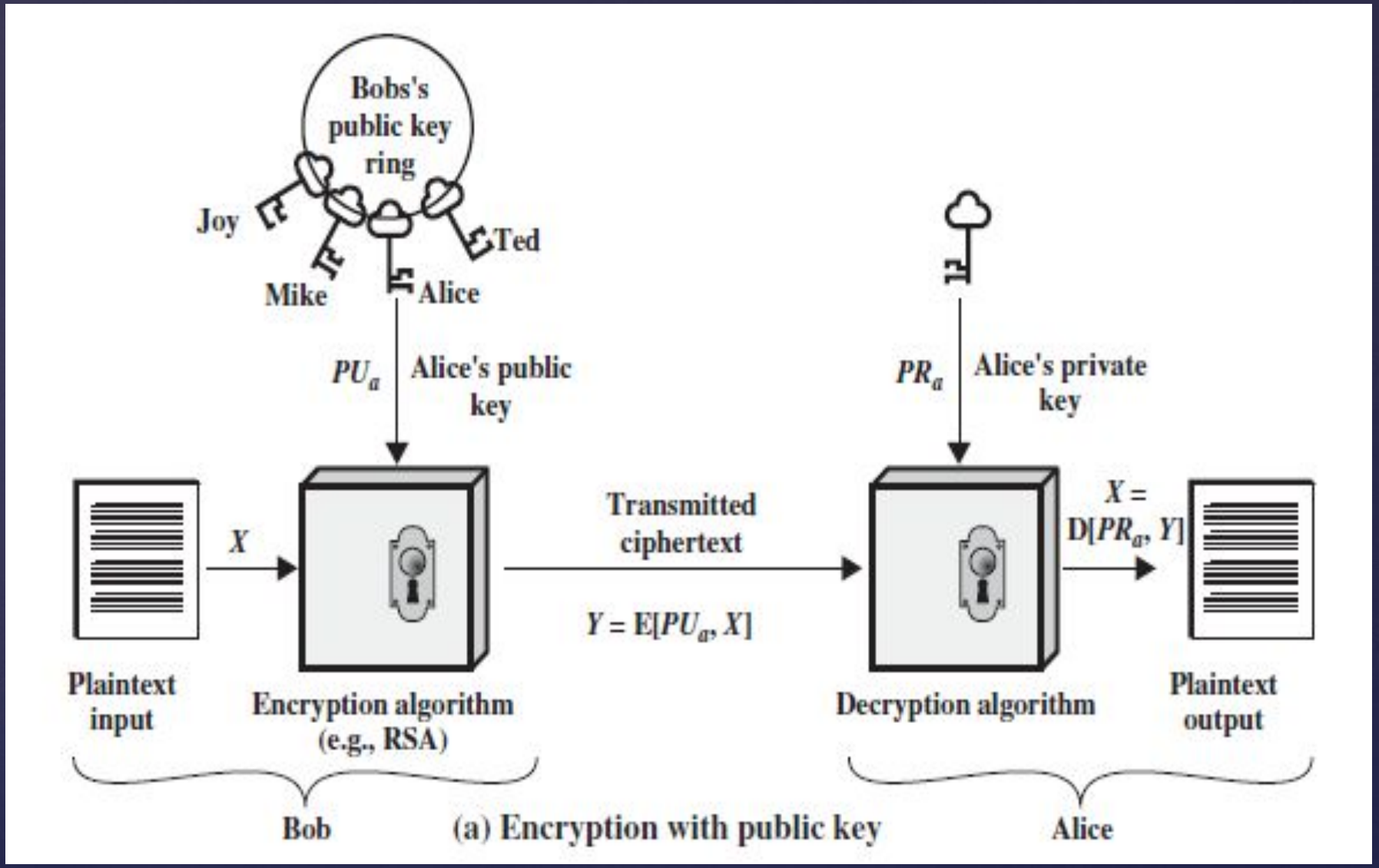
Public-key cryptosystem

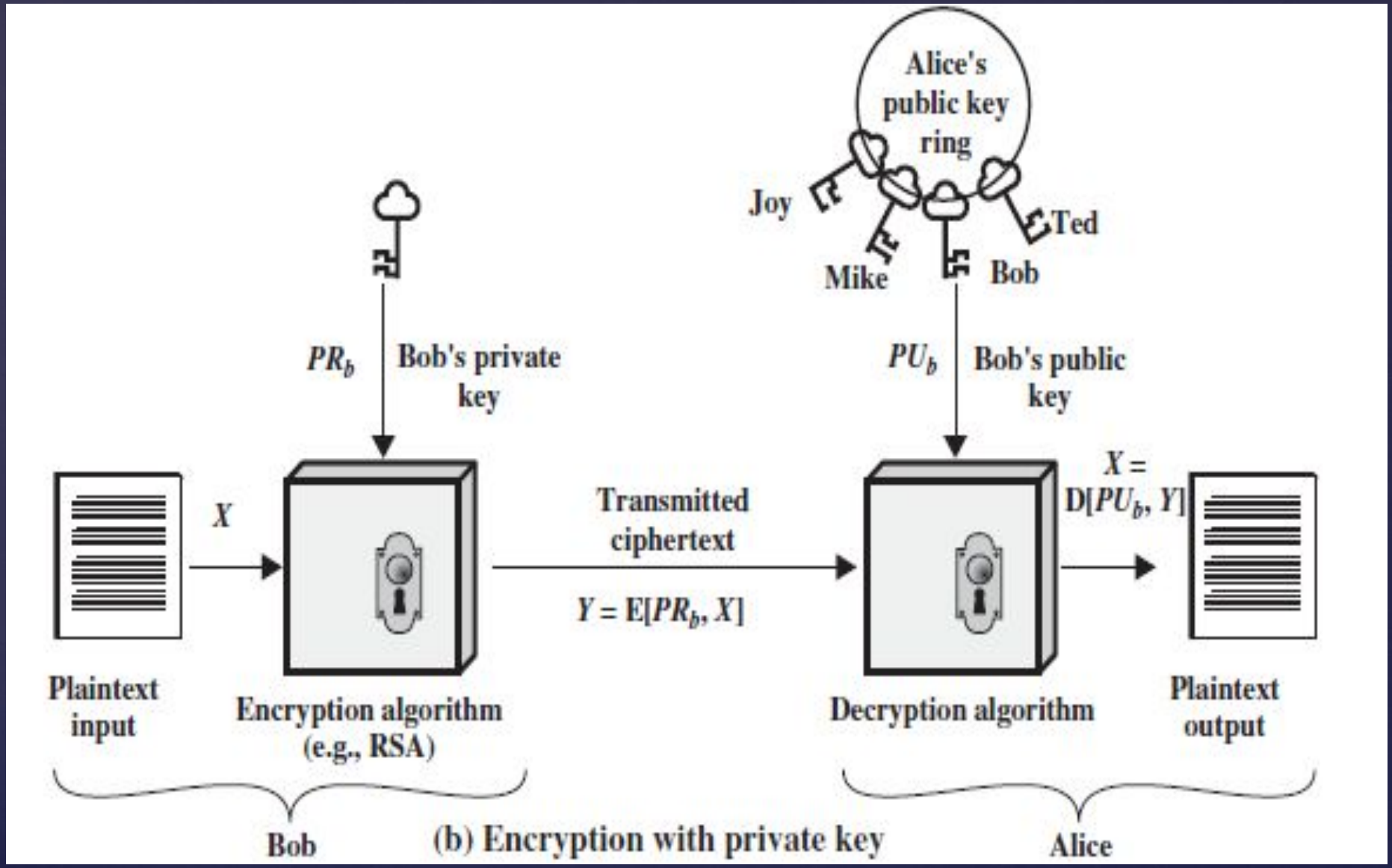
- Asymmetric algorithms rely on one key for encryption and a different but related key for decryption
- These algorithms have the following important characteristics
- It is computationally infeasible to determine the decryption key given only knowledge of the cryptographic algorithm and the encryption key

Six ingredients

- **Plaintext**
  - This is a readable message or data that is fed into the algorithm as the input
- **Encryption algorithm**
  - The encryption algorithm performs various transformations on the plaintext
- **Public and private keys**
  - This is a pair of keys that have been selected so that if one is used for encryption, the other is used for decryption. The exact transformations performed by the algorithm depend on the public key and the private key that is provided as input

- **Cipher text**
  - This is the scrambled message produced as output
- **Decryption algorithm**
  - The algorithm that accepts the cipher text and matching key and produces the original plain text


Bobs's public key ring
Joy
Ted
Mike
Alice
PU_a
Alice's public key
PR_a
Alice's private key
X
Transmitted ciphertext
$Y = E[PU_a, X]$
$X = D[PR_a, Y]$
Plaintext input
Encryption algorithm (e.g., RSA)
Decryption algorithm
Plaintext output
Bob
(a) Encryption with public key
Alice



(b) Encryption with private key

1. Each user generates a pair of keys to be used for the encryption and decryption of messages
2. Each user places one of the two keys in public register or other accessible file. This is public key. The other key is kept private. Each user maintains a collection of public keys obtained from others
3. If Bob wishes to send a confidential message to Alice, Bob encrypts the message using Alice's public key
4. When Alice receives the message, she decrypts it using her private key
5. No other recipient can decrypt the message because only Alice knows her private key

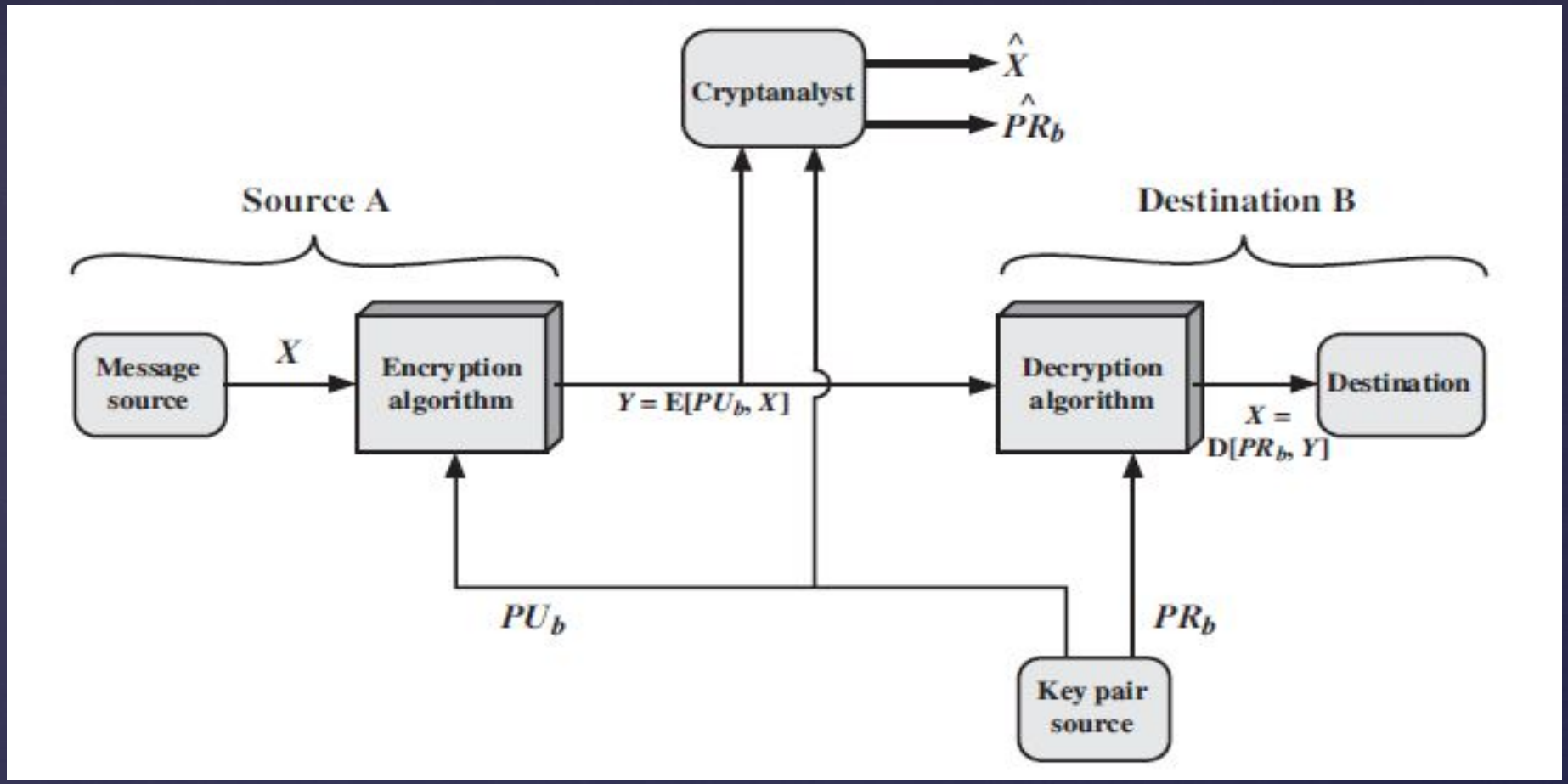
The essential steps

Key points

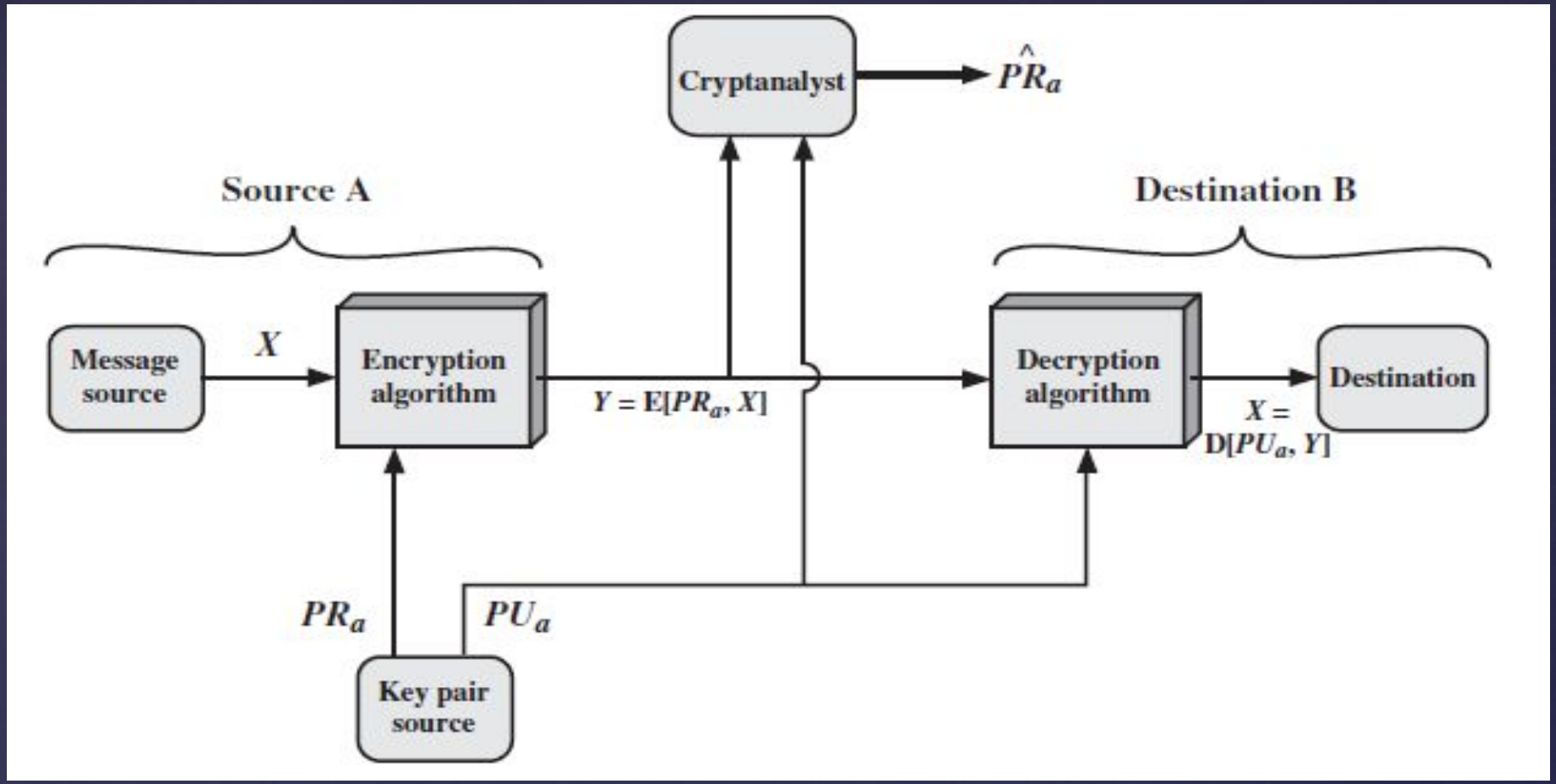
- Here, all participants have access to public keys, and private keys are generated locally by each participant and therefore need never be distributed
- As long as a user's private key remains protected and secret, incoming communication is secure
- At any time, a system can change its private key and publish the related public key to replace its old public key

Conventional Encryption	Public-Key Encryption
Needed to Work: 1. The same algorithm with the same key is used for encryption and decryption. 2. The sender and receiver must share the algorithm and the key. *Needed for Security:* 1. The key must be kept secret. 2. It must be impossible or at least impractical to decipher a message if no other information is available. 3. Knowledge of the algorithm plus samples of ciphertext must be insufficient to determine the key.	*Needed to Work:* 1. One algorithm is used for encryption and decryption with a pair of keys, one for encryption and one for decryption. 2. The sender and receiver must each have one of the matched pair of keys (not the same one). *Needed for Security:* 1. One of the two keys must be kept secret. 2. It must be impossible or at least impractical to decipher a message if no other information is available. 3. Knowledge of the algorithm plus one of the keys plus samples of ciphertext must be insufficient to determine the other key.

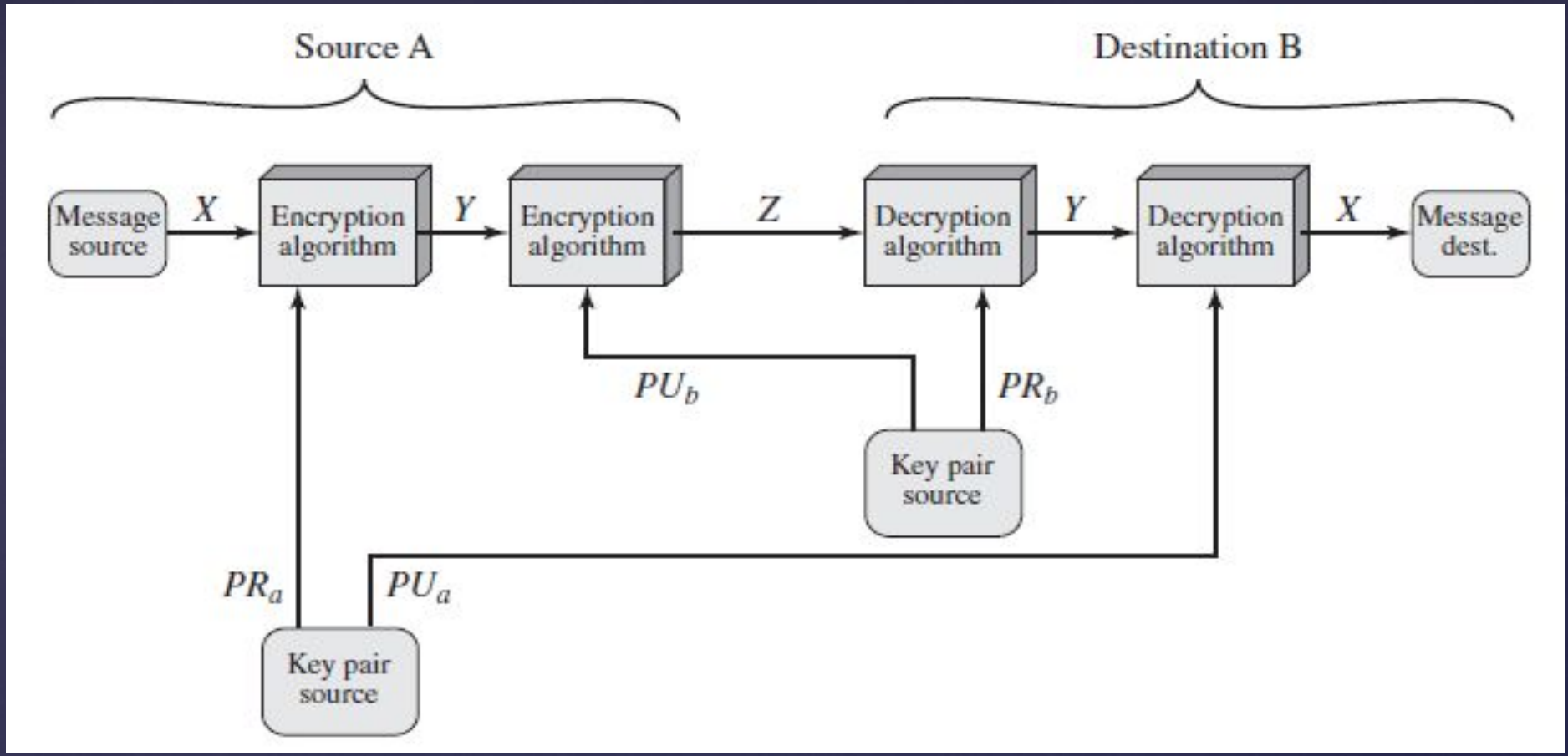
Comparison



Public-key Cryptosystem: Secrecy



Public-key Cryptosystem: Authentication



Public-key Cryptosystem: Authentication and Secrecy

Applications of Public-key Cryptosystem

- Three broad categories:
- Encryption/decryption – The sender encrypts the message with the recipient's public key
- Digital Signature – The sender "signs" a message with its private key
- Key exchange – two sides cooperate to exchange a session key

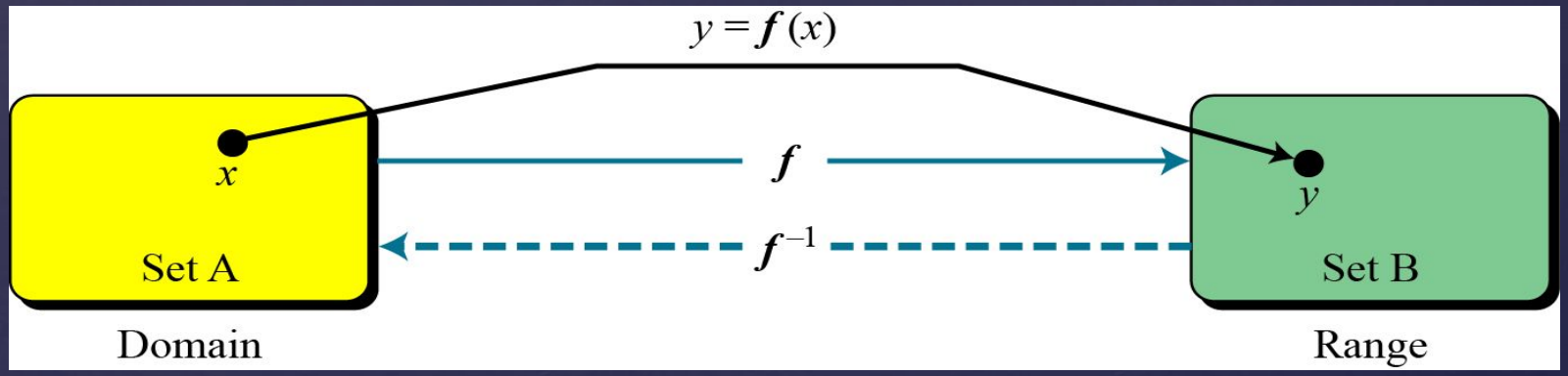
Requirement of Public Key Cryptosystem

- Conditions to be fulfilled by algorithms:
  - **It is computationally easy for party B to generate a key pair(PU_B,PR_B)**
  - **It is computationally easy for a sender, knowing the public key and message to be encrypted to generate the corresponding ciphertext:**

    $$C= E(PU_B,M)$$
  - **It is computationally easy for the receiver to decrypt the resulting ciphertext using the private key to recover the original message:**

    $$M=D(PR_B,C)=D[PR_B,E(PU_B,M)]$$
  - **It is computationally infeasible for an adversary, knowing the public key, PU_B, to determine the private key, PR_B.**
  - **It is computationally infeasible for an adversary, knowing the public key PU_B , and a ciphertext C, to recover original message M.**
  - **Optional: The two keys can be applied in either order.**

- This is the main idea behind Asymmetric key cryptography
- *Function*
  - It is a rule that maps one element in set A(domain) to one element in set B(range)



Trapdoor One-Way function

- *One-Way Function (OWF)*

1. f is easy to compute. For given x, y=f(x).
2. f^{-1} is difficult to compute(infeasible to calculate $x=f^{-1}(x)$.

- Here easy to compute means problem can be solved in polynomial time as a function of input length.
  - Ex: if the length of input is n bits, then the time to compute the function is proportional to $\mathbf{n^a}$.
- The term infeasible means the efforts to solve a problem grows faster than polynomial time as a function of input size.
  - Ex: if the length of input is n bits then the time to compute the function is proportional to $\mathbf{2^n}$.

Trapdoor One-Way function

Trapdoor One-Way function

- *Trapdoor One-Way Function (TOWF)*
  - *It is a One-way function with third property*

*3. Given **y** and a **trapdoor**(secret key), x can be computed easily.*

Trapdoor One-Way function

- Example 1:
  - When n is large, $n = p \times q$ is a one-way function. Given p and q , it is always easy to calculate n ; given n, it is very difficult to compute p and q. This is the factorization problem.
- Example 2:
  - When n is large, the function $y = x^k \bmod n$ is a trapdoor one-way function. Given x, k, and n, it is easy to calculate y. Given y, k, and n, it is very difficult to calculate x. This is the discrete logarithm problem.

RSA Algorithm

- **Asymmetric key cryptographic algorithm**
- Rivest-Shamir-Adleman (RSA) name is given by taking the first name of its inventors
- It uses prime numbers
- This algorithm is based on the fact that it is easy to find and multiply large prime numbers together, but it is extremely difficult to factor their product
- The private and public keys in RSA are based on very large prime numbers
- The real challenge in RSA is the selection and generation of the public key and private key
- Lets know how private key and public key are generated and, using them, how can we perform encryption and decryption

Algorithm

1. Choose two prime numbers P and Q
2. Calculate $N = P * Q$
3. Calculate $\Phi = (P-1) * (Q-1)$
4. Select the public key E (i.e. Encryption key) such that it is not a factor of Φ and $1<E<\Phi$
5. Select the private key D (i.e. Decryption key) such that the following equation is true

   $(D * E) \bmod \Phi = 1$ (Note: Use extended Euclidian algorithm for finding D)
6. For encryption, calculate the cipher text CT from the plain text PT as follows

   $CT = PT^E \bmod N$
7. Send CT as the cipher text to the receiver
8. For decryption, calculate the plain text PT from the cipher text CT as follows

   $PT = CT^D \bmod N$

Example of RSA

1. **Choose two large prime numbers P and Q**
   - Let P= 7, Q=17
2. **Calculate N = P * Q**
   - N= 7 * 17= 119
3. **Calculate Φ = (P–1) * (Q–1) = 96**

4. **Select the public key E such that it is not a factor of 96**
   - The factors of 96 are 2,2,2,2,2 and 3 (because 96 = 2*2*2*2*2*3)
   - Thus we have to choose E such that none of the factors of E is 2 and 3
   - Lets choose E as 5

5. Select the private key D such that the following equation is true

(D * E) mod Φ = 1

- Lets substitute the values of E, P and Q in the equation
- We have (D * 5) mod 96 = 1
- Using Extended Euclidian algorithm, D=77
- So that (77 * 5) mod (96) = 385 mod 96 = 1

6. **For encryption, calculate the cipher text CT from the plain text PT as follows**

$CT = PT^E \bmod N$

- Lets assume that plaintext PT = 10
- Then, $CT = 10^5 \bmod 119 = 100000 \bmod 119 = 40$

7. **Send CT as the cipher text to the receiver**
   - Send 40 as the cipher text to the receiver
8. **For decryption, calculate the plain text PT from the cipher text CT as follows**

$PT = CT^D \bmod N$

- $PT = 40^{77} \bmod 119 = 10$

If $n = p \times q$, $a < n$, and k is an integer, then $a^{k \times \phi(n)+1} \equiv a \pmod{n}$.

$P_1 = C^d \bmod n = (P^e \bmod n)^d \bmod n = P^{ed} \bmod n$

$ed = k\phi(n) + 1$ // d and e are inverses modulo $\phi(n)$

$P_1 = P^{ed} \bmod n \rightarrow P_1 = P^{k\phi(n)+1} \bmod n$

$P_1 = P^{k\phi(n)+1} \bmod n = P \bmod n$ // Euler's theorem (second version)

Proof of RSA

Attacks on RSA

- Four possible approaches to attacking the RSA:
- **Brute force:** This involves trying all possible private keys.
- **Mathematical attacks (Factorization)**: There are several approaches, all equivalent in effort to factoring the product of two primes.
- **Timing attacks**: These depend on the running time of the decryption algorithm.
- **Chosen cipher text attacks:** This type of attack exploits properties of the RSA algorithm.

Attacks on RSA

- Factorization attack
  - The security of RSA is based on the idea that the modulus is so large that it is infeasible to factor it
  - Bob selects p and q and calculates n = p*q
  - Although n is public, p and r are secret
  - If Eve (intruder) can factor n, and obtain p and q, she can calculate Φ(n)
  - Eve can then calculate D as E is public
  - So Eve can easily calculate palintext

Attacks on RSA

- Chosen-ciphertext attack
  - This attack is based on multiplicative property of RSA
  - Assume that Alice Creates ciphertext $C = P^e \bmod n$ and sends it to Bob
  - Also assume that Bob will decrypt an arbitrary ciphertext for Eve rather than C
  - Eve intercepts C and uses the following steps
    - Eve choses a random integer X
    - Eve calculates $Y = C * X^e \bmod n$
    - Eve sends Y to Bob for decryption and gets $Z = Y^d \bmod n$
    - Eve can easily find P because
      - $Z = Y^d \bmod n = (C * X^e)^d \bmod n = = (C^d * X^{ed}) \bmod n = (C^d * X) \bmod n$
      - $Z = (P * X) \bmod n \Rightarrow P = Z * X^{-1} \bmod n$
    - Eve uses extended Euclidian algorithm to find multiplicative inverse of X

References

- Cryptography and Network Security by William Stallings
- Cryptography and Network Security by Behrouz Forouzan